BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS
- BLANK PAGES

IMAGES ARE BEST AVAILABLE COPY.

As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号

特開平10－198643

(43)公開日　平成10年(1998)７月31日

(51)Int.Cl.[6]	識別記号	ＦＩ	
Ｇ０６Ｆ　15/16	３７０	Ｇ０６Ｆ　15/16	３７０Ｎ
		9/46	３６０Ｃ
9/46	３６０	15/16	４２０Ｊ

審査請求　未請求　請求項の数６　ＯＬ　（全 11 頁）

(21)出願番号　特願平9－2061

(22)出願日　平成９年(1997)１月９日

(71)出願人　000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目６番地
(71)出願人　000153443
株式会社日立情報制御システム
茨城県日立市大みか町５丁目２番１号
(72)発明者　加藤　裕昭
茨城県日立市大みか町五丁目２番１号　株式会社日立情報制御システム内
(72)発明者　仲田　智
茨城県日立市大みか町五丁目２番１号　株式会社日立情報制御システム内
(74)代理人　弁理士　高崎　芳紘

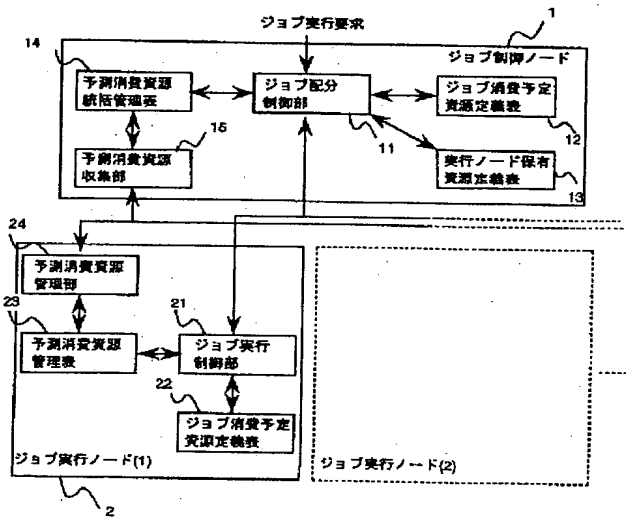
最終頁に続く

(54)【発明の名称】　分散計算機システム

(57)【要約】

【課題】　ジョブを配分するに際し、その配分の判断が計算機資源のジョブ起動時点における情報のみによって行われており、時間の経過に伴い計算機上で実行されている複数のジョブによって資源が消費され、資源の不足により計算機を停止させる可能性がある。

【解決手段】　ジョブ消費予定資源定義表１２に予め各ジョブの消費予定資源が定義されている。ジョブの実行時に、ジョブ配分制御部１１、ジョブ実行制御部２１、ジョブ消費予定資源定義表２２、予測消費資源管理表２３を用い、ジョブ消費予定資源定義表１２の消費予定資源及び予測消費資源情報を基に、前記ジョブを実行するジョブ実行ノード２において必要とされる将来の消費予定資源が前記ノードの有する資源をこえない様にジョブを配分し、稼働停止を防止する。



1

【特許請求の範囲】

【請求項１】　複数の計算機から構成された分散計算機システムにおいて、実行対象のジョブの消費予定資源を予め定義するジョブ消費予定資源定義手段と、前記ジョブの実行時に、そのジョブを実行するノード上で必要とされる将来の消費予定資源が前記ノードの有する資源を超えない様にジョブの配分を行うジョブ配分手段とを具備することを特徴とする分散計算機システム。

【請求項２】　前記ジョブ配分手段は、前記ジョブ消費予定資源定義手段で定義されたジョブの消費予定資源と予測消費資源統括管理手段で管理されている前記ノード上の予測消費資源との和がノード保有資源定義手段で定義された前記ノード上の保有資源を越えないようにジョブを配分することを特徴とする請求項１記載の分散計算機システム。

【請求項３】　前記ジョブ配分手段は、前記ジョブ配分に加え、前記ジョブ消費予定資源定義手段で定義されたジョブの消費予定資源と予測消費資源統括管理手段で管理されている前記ノード上の予測消費資源との和に対する前記ノード上の予測消費資源の比率が最小になり、かつ前記和がノード保有資源定義手段で定義された前記ノード上の保有資源を越えないようにジョブを配分することを特徴とする請求項２記載の分散計算機システム。

【請求項４】　前記ジョブ配分手段は、スキップされたジョブの起動回数が予め設定した値を越える時、前記スキップされたジョブを優先的に実行させることを特徴とする請求項１記載の分散計算機システム。

【請求項５】　前記ノードに代えて、データベースサーバを用いることを特徴とする請求項１記載の分散計算機システム。

【請求項６】　前記ノード又は前記データ処理サーバは、ジョブ実行要求に応じて消費予定資源値又は記憶資源アカウント値を取得し、この値を予測消費資源管理手段又は記憶資源アカウントテーブルに加算してジョブを起動し、その実行後に前記予測消費資源管理手段から消費予定資源値を減算又は前記記憶資源アカウントテーブルから前記記憶資源アカウント値を減算する手段を具備することを特徴とする請求項１又は請求項５記載の分散計算機システム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、複数の計算機ノードで構成された分散計算機システムに関するものである。

【０００２】

【従来の技術】近年の計算機システムは、処理を高速化する要求、処理データ量の増大、及び計算機間を接続するネットワークの発展により、複数の計算機をネットワークで接続した分散計算機システムを構成するケースが多くなってきている。分散計算機システム上でジョブを

2

実行する場合、計算機資源（１群の仕事を計算機システムで実行するために必要なハードウェア及びソフトウェア）を有効に活用するためには、個々の計算機の資源消費状態を管理しながら各計算機に最適にジョブを配分する必要がある。

【０００３】従来、分散計算機システムにおけるジョブ配分は、ジョブの新規起動時に、（ｉ）各計算機のジョブ配分時点における資源消費状態、及び（ii）予め定められたポリシーに基づいてジョブの配分を決定していた。

【０００４】

【発明が解決しようとする課題】しかし、従来の分散計算機システムにおけるジョブ配分は、ジョブの制御を行うために管理する各計算機上の資源情報が、刻々変化する計算機上の資源のある時間断面におけるデータに基づいている。このため、ジョブの起動時には計算機の空き資源がジョブを実行するに十分と判断されているにもかかわらず、時間の経過に伴い計算機上で実行されている複数のジョブによって資源が消費され、計算機を資源の不足により停止させてしまう可能性がある。

【０００５】なお、計算機の資源が予め設定したしきい値を超えて消費された場合、アラーム等によりオペレータに通知し、オペレータがジョブのキャンセル等の措置を行うことにより、計算機の停止を未然に防ぐことは可能である。しかし、実行を中止したジョブを他の計算機で再起動する必要があること、及び計算機が終日運転する場合、オペレータが常時監視しなければならないという不便さがある。これは、ジョブ配分の判断が、計算機資源のジョブ起動時点における情報のみによって行われていることに理由があり、これを解決するためには、計算機上で実行されているすべてのジョブがそのライフタイム（ジョブの起動から停止までの時間）にわたって消費する資源を管理する必要がある。

【０００６】本発明の目的は、ノードの稼働停止を防止することのできる分散計算機システムを提供することにある。

【０００７】

【課題を解決するための手段】上記の目的を達成するために、本発明は、複数の計算機から構成された分散計算機システムにおいて、実行対象のジョブの消費予定資源を予め定義するジョブ消費予定資源定義手段と、前記ジョブの実行時に、そのジョブを実行するノード上で必要とされる将来の消費予定資源が前記ノードの有する資源を超えない様にジョブの配分を行うジョブ配分手段とを備えた構成にしている。この構成によれば、予め各ジョブの使用予定資源をジョブ消費予定資源定義手段に定義しておき、ジョブの実行時に、個々のノードに一度にジョブが配分されないような配分、つまりジョブ配分を各ノードの有する資源を超えないように決定すれば、ジョブの起動に際して該ジョブを実行するために必要な計算

機上の資源が、ジョブのライフタイムにわたって予測消費資源として仮想的に確保される。この結果、計算機上で実行されている全てのジョブがそのライフタイムにわたって消費する資源を管理できるようになり、計算機資源が各ジョブの実行に伴って不足する事態を招くことがなくなる。

【０００８】

【発明の実施の形態】以下、本発明の実施の形態について説明する。図１は本発明による分散計算機システムのジョブ配分処理部分を示すブロックを示している。図１には分散計算機システムのジョブ制御ノード１と、このジョブ制御ノード１に接続されたジョブ実行ノード２が示されている。ジョブ制御ノード１は、ユーザからのジョブ実行要求を基に動作するジョブ配分制御部１１、定義されたジョブ消費予定資源をジョブ配分制御部１１に提供するジョブ消費予定資源定義表１２、予め定義された各実行ノードの保有資源情報を保持し、これをジョブ配分制御部１１に提供する実行ノード保有資源定義表１３、ノード上の予測消費資源をジョブ配分制御部１１に提供する予測消費資源統括管理表１４、及びジョブ実行ノード２側から予測消費資源情報を収集して予測消費資源統括管理表１４に反映する予測消費資源収集部１５を備えている。

【０００９】また、ジョブ実行ノード２は、ジョブ制御ノード１側からの実行要求により起動するジョブ実行制御部２１、定義されたジョブ消費予定資源をジョブ実行制御部２１に提供するジョブ消費予定資源定義表２２、ジョブ実行制御部２１により当該実行ノードの予測消費資源が更新される予測消費資源管理表２３、及び予測消費資源管理表２３の内容を参照して予測消費資源を管理する予測消費資源管理部２４を備えている。なお、ジョブ実行ノード２は複数が存在可能であるが、図１では１つのみの構成を示し、その動作説明においても１つのみについて説明する。また、図１においては、ジョブ制御ノード１とジョブ実行ノード２を別個のノードとして構成しているが、ジョブ制御ノード１がジョブ実行ノード２を含む構成であってもよい。

【００１０】以上の構成において、ジョブ配分制御部１１は、ユーザからのジョブ実行要求に対し、予測消費資源統括管理表１４で管理される予測消費資源（ジョブ実行ノード上の）Ｒfcと、ジョブ消費予定資源定義表１２に定義されるジョブの消費予定資源Ｒcpの和（Ｒfc＋Ｒcp）に対する前記Ｒfcの比率Ｒfc／（Ｒfc＋Ｒcp）が最小になり、且つ、前記Ｒfc＋Ｒcpが実行ノード保有資源定義表１３に定義される当該ジョブ実行ノードの保有資源を超えないジョブ実行ノード２を選択する。この選択したジョブ実行ノード２上のジョブ実行制御部２１に対し、ジョブの実行要求が行われる。

【００１１】ジョブ配分制御部１１からジョブの実行要求を受けたジョブ実行制御部２１は、そのジョブを起動すると共にジョブ消費予定資源定義表２２から前記ジョブの消費予定資源値を取得する。この消費予定資源値は、ジョブ実行制御部２１によって予測消費資源管理表２３に加算され、また、ジョブの実行が終了すれば、前記ジョブの消費予定資源値は予測消費資源管理表２３から減算され、予測消費資源の更新が行われ、予測消費資源の状態が管理される。この後、予測消費資源管理部２４によって予測消費資源管理表２３の参照が行われ、この参照結果を基に予測消費資源情報が取得され、この予測消費資源情報は予測消費資源管理部２４から予測消費資源収集部１５へ送出される。

【００１２】このように、予め各ジョブの使用予定資源をジョブ消費予定資源定義表１２、２２に定義しておき、ジョブの実行時に、各ノードにジョブが一度に配分されないようにジョブ配分制御部１１によって決定すれば、計算機資源を有効に活用できる。この結果、計算機上で実行されている全てのジョブがそのライフタイムにわたって消費する資源を管理でき、計算機資源が各ジョブの実行に伴って不足する事態は生じなくなる。

【００１３】本発明は、特に、画像処理や科学技術計算分野のように大量の資源を消費する類似のジョブを多数実行する等の用途に有効である。つまり、各計算機に一度にジョブを配分し過ぎることによって計算機に許容以上の資源が要求され、計算機の稼働に悪影響を与えるようなシステムへの適用に有効である。

【００１４】図２は図１に示したジョブ配分制御部１１の詳細を示すブロック図である。ジョブ配分制御部１１は、ジョブ実行要求管理部１６、ジョブ実行要求キュー管理表１７、及びジョブ配分処理部１８を備えて構成されている。ジョブ実行要求管理部１６は、ユーザからジョブ実行要求を受けたときにジョブ実行要求をジョブ実行要求キュー管理表１７のキューに登録する。また、ジョブ配分処理部１８は、予め定義された周期毎、又はユーザからジョブ実行要求があった時ジョブ実行要求管理部１６からの起動指令に基づいて、ジョブ実行要求キュー管理表１７からジョブ実行要求を取り出し、ジョブ消費予定資源定義表１２、実行ノード保有資源定義表１３及び予測消費資源統括管理表１４の３つを参照して各ジョブのジョブ実行ノードへの配分を実行する。

【００１５】更に、ジョブ配分処理部１８は、どのジョブ実行ノード上に対しても必要とする予測消費資源を確保できなかったジョブの配分を見合わせ、ジョブ実行要求キュー管理表１７内の次の順番のジョブを配分することを試みる。これは、ジョブ実行ノードの効率的な使用を実現するものである。しかし、このようなジョブの配分を行っていると、大量の消費予定資源を必要とするジョブが、いつまでもジョブ実行ノード２に配分されないケースが発生する。そこで、ジョブ配分処理部１８は、ジョブの配分をスキップする度にジョブ実行要求キュー管理表１７内の当該ジョブのスキップカウンタ（図示せ

ず）をインクリメントする。そして、カウント値が予め定義（スキップカウンタに）したしきい値に達した場合、前記ジョブが優先的に実行されるように処理する。このジョブ配分方式にすることで、ジョブに対する公平な資源配分の実現、すなわちジョブを公平に実行することが可能になる。なお、ジョブ消費予定資源定義表２２は、ジョブ配分制御部１１がジョブ実行制御部２１に対してジョブの実行要求を行う時に、当該ジョブの消費予定資源を合わせて通知することにより削除することも可能である。

【００１６】次に、本発明による分散計算機システムの具体例について説明する。図３は、大量の入力データに処理及び加工を施して出力するジョブを複数のデータ処理サーバ上で実行する分散計算機システムのブロック図を示している。入力データのサイズは、ユーザから与えられるジョブ実行要求毎に異なるので、各ジョブが出力するデータのサイズもジョブ実行要求毎に異なる。このため、従来のジョブ配分技術では、データ処理サーバ上で各ジョブの実行が進むにつれ、出力データを格納するディスク（記憶媒体）を使いきってしまう可能性があった。ディスクに出力データを格納できないということは、データ処理サーバの稼働停止を意味する。したがって、かかる事態はジョブ配分の過程で避ける必要があった。しかし、本発明の適用により、各データ処理サーバ上で実行中のジョブがそのライフタイムにわたって消費予定のディスク容量を把握でき、その情報に基づいたジョブ配分を行うことが可能になるので、データ処理サーバの稼働停止の危険性を著しく低下させることができる。

【００１７】以下、図３の構成及び動作について説明する。図３には管理エージェント１０１と、複数のデータ処理サーバ２０１の構成が示され、管理エージェント１０１は図１のジョブ制御ノード１に相当し、データ処理サーバ２０１は図１のジョブ実行ノード２に相当する。管理エージェント１０１は、ジョブ属性定義ファイル１２１、ジョブ属性ジェネレータ１２２、図１のジョブ消費予定資源定義表１２に相当するジョブ属性定義テーブル１２３、ノード記憶資源定義ファイル１３１、ノード記憶資源ジェネレータ１３２、図１の実行ノード保有資源管理表１３に相当するノード記憶資源定義テーブル１３３、記憶資源統括アカウントテーブル１４１、記憶資源アカウント収集マスタ１５１、図２のジョブ実行要求管理部１６に相当するジョブ実行受付処理部１６１、図２のジョブ実行要求キュー管理表１７に相当するジョブ管理テーブル１７１、図２のジョブ配分処理部１８に相当するジョブディスパッチャ１８１、及びジョブ状態更新処理部１８２を備えて構成されている。

【００１８】また、データ処理サーバ２０１は、図１のジョブ実行制御部２１に相当するジョブ起動処理部２１１、ジョブシェル２１２、記憶資源アカウントテーブル２３１、及び図１の予測消費資源管理部２４に相当する記憶資源アカウント収集スレーブ２４１を備えて構成されている。なお、データ処理サーバ２０１は複数台の存在が可能であるが、以下においてはデータ処理サーバ２０１の１つについてのみ説明する。

【００１９】管理エージェント１０１において、ジョブ属性定義テーブル１２３は管理エージェントの不図示のメモリ上に存在しており、これに対してジョブ属性定義ファイル１２１及びジョブ属性ジェネレータ１２２はその補助機構として機能する。具体的に説明すると、ユーザがジョブ属性定義ファイル１２１にジョブの属性を記述し、ジョブ属性ジェネレータ１２２を実行することによって、ジョブ属性定義ファイル１２１に格納したジョブの属性をメモリ上のジョブ属性定義テーブル１２３に反映することができる。本実施例において、ジョブの属性とは、（１）ジョブ実行要求時にユーザから与えられる入力データのサイズに対してジョブ実行の過程で生成される中間データ及び最終的な出力データのサイズの関係式、（２）上記ジョブを起動する場合のスキップリミット値、を含むものである。

【００２０】ノード記憶資源定義テーブル１３３は管理エージェント１０１のメモリ上に存在し、これに対してノード記憶資源定義ファイル１３１及びノード記憶資源ジェネレータ１３２は、ノード記憶資源定義テーブル１３３の補助機構として機能する。具体的に説明すると、ユーザによりノード記憶資源定義ファイル１３１に各ノードが有する記憶資源量の情報が記述され、ノード記憶資源ジェネレータ１３２が動作することにより、各ノードが有する記憶資源量をメモリ上のノード記憶資源定義テーブル１３３に反映することができる。本実施例において、ノード記憶資源定義ファイル１３１に定義されるノード記憶資源とは、各データ処理サーバが有するディスクの容量を含むものである。

【００２１】ジョブ管理テーブル１７１は、管理エージェント１０１のメモリ上に存在する。また、ジョブディスパッチャ１８１は予め定義された周期で起動し、各データ処理サーバ２０１の記憶資源アカウント情報テーブル２３１の内容の収集を記憶資源アカウント収集マスタ１５１に要求し、その収集結果が格納される記憶資源統括アカウントテーブル１４１を参照してジョブ実行要求をジョブ管理テーブル１７１から取り出し、そのジョブをデータ処理サーバ２０１のいずれかにディスパッチする。

【００２２】ここで、ジョブディスパッチャ１８１は、ジョブを実行するために必要な記憶資源の空きがデータ処理サーバ２０１のいずれにも存在しない場合、そのジョブの実行を見合わせ、ジョブ管理テーブル１７１内のジョブのスキップカウンタをインクリメントする。スキップカウンタの値がジョブ属性定義テーブル１２３から与えられるスキップリミット値に達した場合、以後ジョ

ブディスパッチャ１８１はそのジョブをスキップさせず、データ処理サーバ２０１のいずれかにおいてジョブを実行するのに十分な記憶資源の空きが生じるまで待機する。また、ジョブディスパッチャ１８１はジョブのディスパッチに際し、ジョブ属性定義テーブル１２３から取得した情報を基にジョブディスパッチャ１８１が生成するジョブの記憶資源アカウント値をジョブ起動パラメータとし、これをジョブ起動処理部２１１へ転送する。これにより、図１で必要であったジョブ消費予定資源定義表２２を省略することができる。

【００２３】ジョブ管理テーブル１７１は、ユーザから要求されたジョブのキュー管理情報に限らず、データ処理サーバ２０１上で実行されている各ジョブの状態も、ジョブ要求受付処理部１６１を通して保持する。なお、図３には示していないが、ジョブ管理テーブル１７１は、画面等を通して各ジョブの実行状態をユーザに提供するための元データとして用いることができる。

【００２４】記憶資源統括アカウントテーブル１４１は図１の予測消費資源統括管理表１４に相当し、管理エージェントのメモリ上に存在する。また、記憶資源アカウント収集マスタ１５１は図１の予測消費資源収集部１５に相当する。この記憶資源アカウント収集マスタ１５１は、ジョブディスパッチャ１８１からの要求に従って記憶資源アカウント収集スレーブ２４１から各データ処理サーバ２０１の記憶資源アカウント情報を収集し、記憶資源統括アカウントテーブル１４１へ格納する。記憶資源アカウントテーブル２３１は図１の予測消費資源管理表２３に相当し、データ処理サーバのメモリ上に存在している。また、記憶資源アカウント収集スレーブ２４１は記憶資源アカウント収集マスタ１５１からの要求により、記憶資源アカウント収集テーブル２３１の内容を記憶資源アカウント収集マスタ１５１へ転送する。ジョブシェル２１２はジョブを包含するものであり、図１に示したジョブ実行制御部２１が予測消費資源管理表２３へ予測消費資源を格納する部分の機能と同一の機能を有している。

【００２５】図４はジョブシェル２１２で実行される処理を示すフローチャートである。まず、ジョブディスパッチャ１８１からジョブ起動処理部２１１に対してジョブ起動要求のパラメータとして渡された記憶資源アカウント値を取得する（ステップＳ４０１）。次に、取得した記憶資源アカウント値を記憶資源アカウントテーブル２３１に加算する（ステップＳ４０２）。更に、ジョブが実行され（ステップＳ４０３）、このジョブの終了後、記憶資源アカウントテーブル２３１から記憶資源アカウント値を減算する（ステップＳ４０４）。最後に、ジョブ状態更新処理部１８２に対してジョブの終了通知を実行する（ステップＳ４０５）。このように、簡単な処理により、ジョブの実行に伴い記憶資源統括アカウントテーブル１４１の内容を更新することができる。

【００２６】図５は、ジョブ属性定義テーブル１２３の構成内容を示している。ここではジョブ毎に、消費予定メモリ容量、消費予定ディスク容量及びスキップリミットを定義している。図５において、ジョブ「ＪＯＢ－Ａ」の消費予定ディスク容量は、固定的に使用される５０ＭＢに、入力データサイズの２．２倍を加えて算出（５０＋２．２×入力データサイズ（ＭＢ））される。

【００２７】図６はジョブ管理テーブル１７１の構成内容を示している。図６において、第１エントリのジョブ「Ａ０１」は、消費予定メモリ容量及び消費予定ディスク容量を確保できるデータ処理ホストが現状存在しないために、起動がスキップされていることを示している。この例では、スキップカウンタがスキップリミットに達しているため、ジョブディスパッチャ１８１は、これ以上ジョブ「Ａ０１」の起動をスキップせず、ジョブ「Ａ０１」の実行に必要な消費予定メモリ容量及び消費予定ディスク容量が、データ処理ホストのいずれかに確保できるまで他のジョブの起動要求は行わない。

【００２８】図７はノード記憶資源定義テーブル１３３の構成内容を示している。ここでは、データ処理サーバ毎に、スワップ容量を含めた保有メモリ容量、及び保有ディスク容量を定義している。図８は記憶資源アカウントテーブル２３１の構成内容を示している。本例では、当該データ処理ホスト上で実行されている各ジョブの消費予定メモリ容量及び消費予定ディスク容量が格納されている。

【００２９】図９は記憶資源統括アカウントテーブル１４１の構成内容を示している。この例では、データ処理サーバ毎に、ジョブの消費予定メモリ容量合計及び消費予定ディスク容量合計を格納している。以上のように、データ処理サーバ２０１を備えたシステムにおいても、予め各ジョブの使用予定資源をジョブ属性定義テーブル１２３及びデータ処理サーバ２０１に定義しておくことにより、計算機上で実行されている全てのジョブがそのライフタイムにわたって消費する資源を管理できるようになり、計算機資源が各ジョブの実行に伴って不足する事態は生じない。

【００３０】図１０は図３の構成を発展させた本発明の他の具体例を示すブロック図である。図１０においては、管理エージェント１０１及びデータ処理サーバ２０１で共有できる情報が、データベースサーバ３０１で管理されるシステム構成が示されている。データベースサーバ３０１上における情報の共有は、ＮＦＳ（網ファイルシステム：Network File/System）等によるファイルの共有、ＳＱＬ（構造化照会言語：Structured Query Language）等を使用したＤＢＭＳ（データベース管理システム： Data Base Management System）等によるものである。

【００３１】図１０の分散計算機システムの構成について説明すると、図１のジョブ制御ノードに相当する管理

エージェント１０１、複数のデータ処理サーバ２０１、及び管理エージェント１０１とデータ処理サーバ２０１に接続されるデータベースサーバ３０１を備えて構成されている。ここでは、三者を独立した形で構成しているが、データベースサーバ３０１が管理エージェント１０１又はデータ処理サーバ２０１を兼ねた構成にすることもできる。

【００３２】管理エージェント１０１は、図２のジョブ実行要求管理部１６に相当するジョブ実行受付処理部１６１と図２のジョブ配分処理部１８に相当するジョブディスパッチャ１８１を備えて構成されている。また、データ処理サーバ２０１は、図１のジョブ実行制御部２１に相当するジョブ起動処理部２１１、及びジョブシェル２１２を備えて構成されている。また、データベースサーバ３０１は、図２のジョブ実行要求キュー管理表１７に相当するジョブ管理テーブル３１１、図１のジョブ消費予定資源定義表１２に相当するジョブ属性定義テーブル３１２、図１の実行ノード保有資源管理表１３に相当するノード記憶資源定義テーブル３１３、及び図１の予測消費資源統括管理表１４に相当する記憶資源統括アカウントテーブル３１４を備えて構成されている。

【００３３】管理エージェント１０１のジョブ実行受付処理部１６１は、ユーザからのジョブ実行要求を受け付け、データベースサーバ３０１内のジョブ管理テーブル３１１にジョブ実行要求のエントリを追加する。ジョブディスパッチャ１８１は予め定義された周期で起動し、ジョブ管理テーブル３１１からジョブ実行要求を取り出し、ジョブ属性定義テーブル３１２に定義されたジョブの予測消費資源情報、ノード記憶資源定義テーブル３１３に定義された各データ処理サーバの保有記憶資源情報、及び記憶資源統括アカウントテーブル３１４で管理されている各データ処理サーバの記憶資源消費情報の各々に基づいて、ジョブを実行するデータ処理サーバを決定する。更に、ジョブディスパッチャ１８１は、ジョブ属性定義テーブル３１２に定義される情報に基づいて算出したジョブの記憶資源アカウント値をパラメータとして、ジョブ起動処理部２１１へジョブの起動要求を送出する。この要求に従って、ジョブ起動処理部２１１はジョブシェル２１２を起動させる。このジョブシェル２１２の処理について、以下に説明する。

【００３４】図１１はジョブシェル２１２の処理を示すフローチャートである。まず、ジョブディスパッチャ１８１からジョブ起動処理部２１１へジョブ起動要求のパラメータとして渡された記憶資源アカウント値を取得する（ステップＳ６０１）。次に、この記憶資源アカウント値を記憶資源統括アカウントテーブル３１４に加算する（ステップＳ６０２）。更に、起動させたジョブを実行し（ステップＳ６０３）、このジョブの実行が終了した後、記憶資源統括アカウントテーブル３１４の記憶資源アカウント値を減算する（ステップＳ６０４）。ついで、ジョブ管理テーブル３１１にジョブの終了ステータスを反映させる（ステップ６０５）。図１１の処理により、ジョブの実行に伴い記憶資源統括アカウントテーブル３１４の内容を更新することができる。

【００３５】以上のように、図１０の構成によれば、管理エージェント１０１及びデータ処理サーバ２０１で共有できる情報がデータベースサーバ３０１で管理されるシステムにおいても、図３の場合と同様に、予め各ジョブの使用予定資源をジョブ属性定義テーブル３１２に定義し、これを用いて適正配分を行うことにより、ジョブの起動に際して該ジョブを実行するために必要な計算機上の資源をジョブのライフタイムにわたって予測消費資源として仮想的に確保でき、計算機上で実行されている全てのジョブがそのライフタイムにわたって消費する資源を管理できるようになり、計算機資源が各ジョブの実行に伴って不足する事態は生じない。

【００３６】

【発明の効果】以上説明した通り、本発明によれば、計算機システムにおいて、実行するジョブの消費予定資源を予めジョブ属性として定義しておくことにより、ジョブの起動時に、ノード上に必要とされる将来の消費予定資源が、ノードが有している資源を超えないようなジョブの配分が行え、ジョブ実行ノードの稼働停止の発生を著しく低減した分散計算機システムを実現することができる。

【図面の簡単な説明】

【図１】本発明による分散計算機システムのジョブ配分処理部分を示すブロック図である。

【図２】図１のジョブ配分制御部の詳細を示すブロック図である。

【図３】本発明の第１の具体例の構成を示すブロック図である。

【図４】図３のジョブシェルの処理を説明するフローチャートである。

【図５】ジョブ属性定義テーブル１２３の構成例図である。

【図６】ジョブ管理テーブル１７１の構成例図である。

【図７】ノード記憶資源定義テーブル１３３の構成例図である。

【図８】記憶資源アカウントテーブル２３１の構成例図である。

【図９】記憶資源統括アカウントテーブル１４１の構成例図である。

【図１０】本発明の第２の具体例の構成を示すブロック図である。

【図１１】図１０のジョブシェルの処理を説明するフローチャートである。

【符号の説明】

１　ジョブ制御ノード

２　ジョブ実行ノード

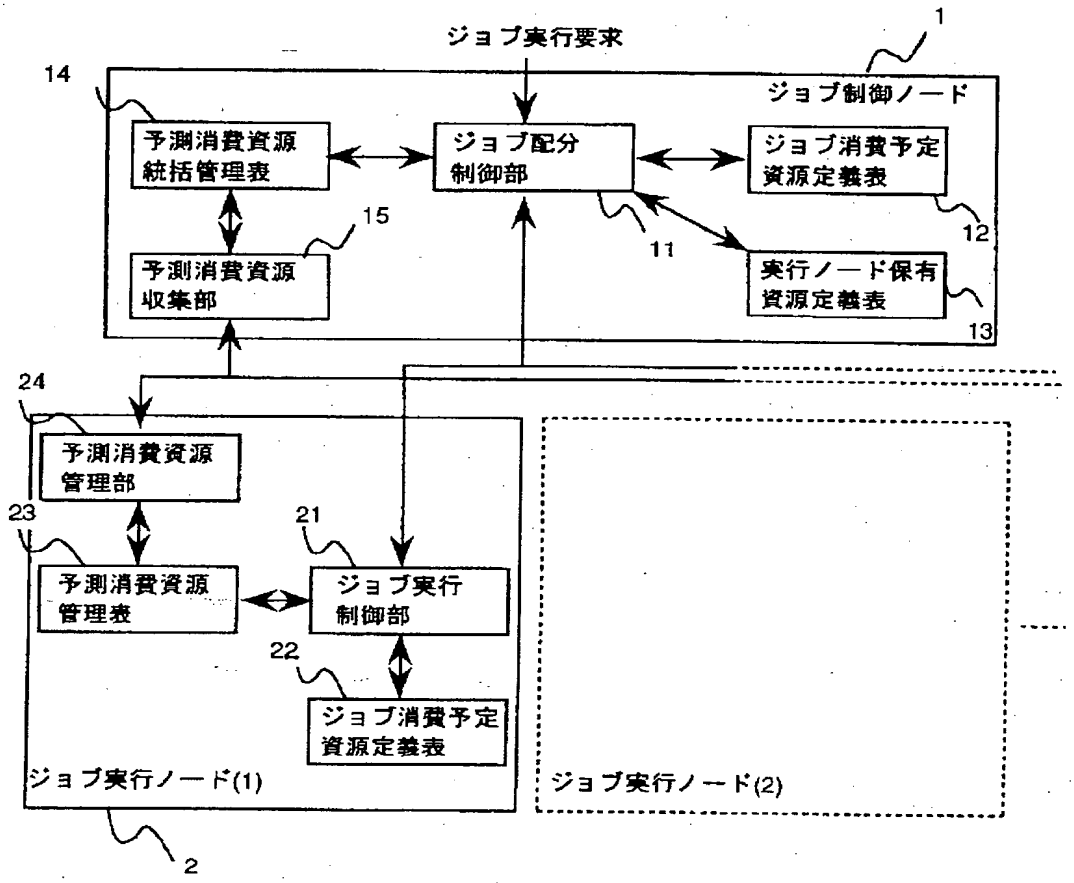
11

１１　ジョブ配分制御部
１２　ジョブ消費予定資源定義表
１３　実行ノード保有資源管理表
１４　予測消費資源統括管理表
１５　予測消費資源収集部
１６　ジョブ実行要求管理部
１７　ジョブ実行要求キュー管理表
１８　ジョブ配分処理部
２１　ジョブ実行制御部
２２　ジョブ消費予定資源定義表
２３　予測消費資源管理表

12

２４　予測消費資源管理部
１０１　管理エージェント
１２３、３１２　ジョブ属性定義テーブル
１３３、３１３　ノード記憶資源定義テーブル
１４１、３１４　記憶資源統括アカウントテーブル
１８１　ジョブディスパッチャ
２０１　データ処理サーバ
２１１　ジョブ起動処理部
２１２　ジョブシェル
３０１　データベースサーバ

【図１】

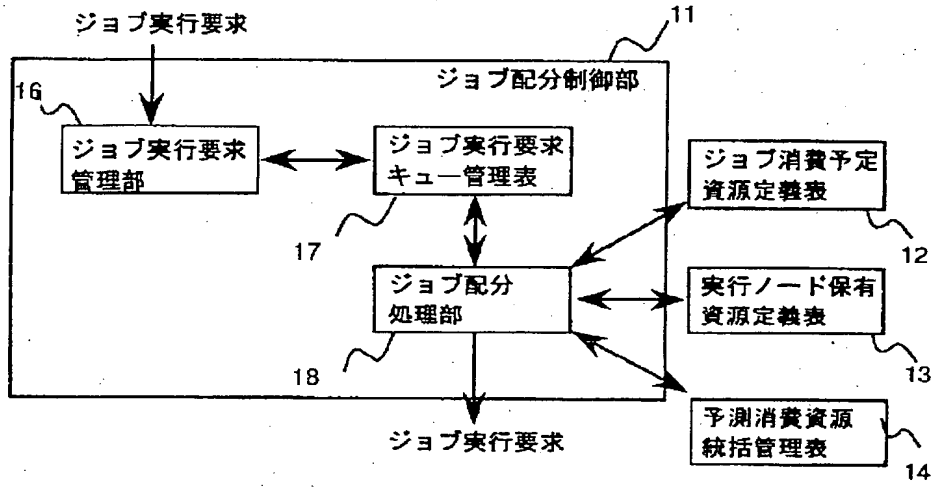


【図８】

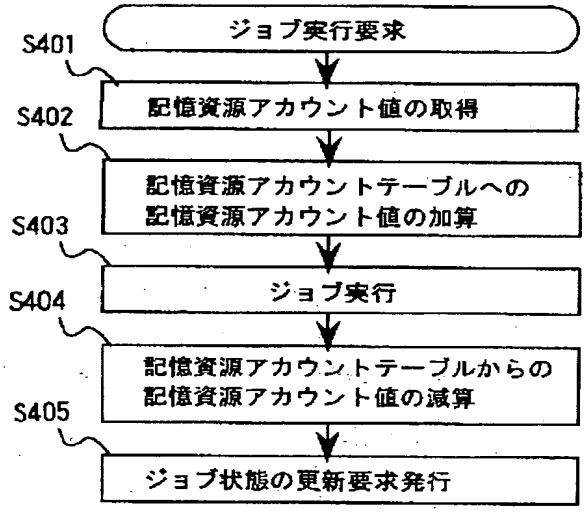
テーブル２３１

ジョブＩＤ	消費予定メモリ容量（ＭＢ）	消費予定ディスク容量（ＭＢ）
Ｂ０１	５	３０

【図２】



【図４】



【図７】

テーブル１３３

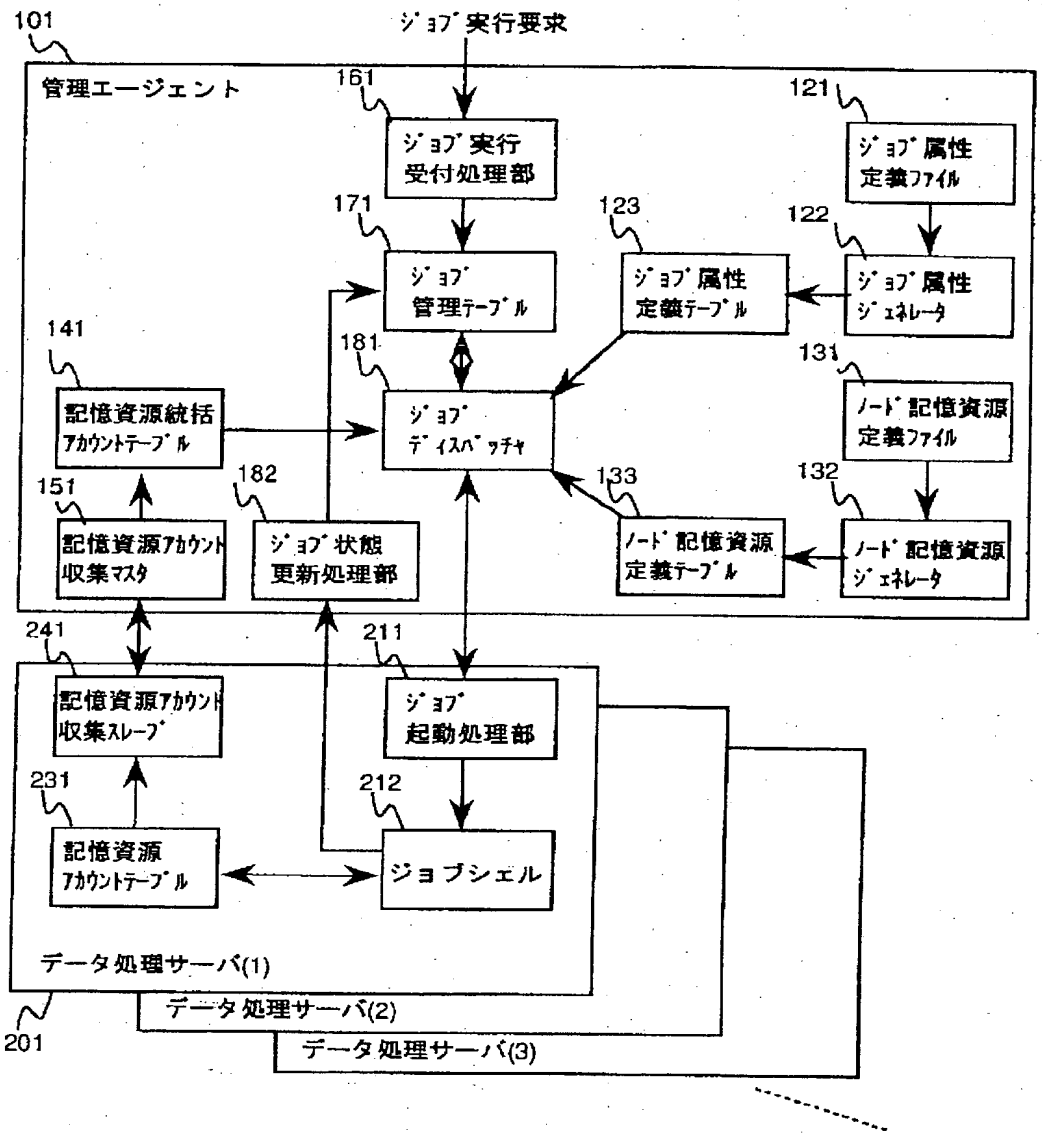
ノードＩＤ	保有メモリ（ＭＢ）	保有ディスク容量（ＭＢ）
ＳＲＶ０１	２，０４８	３０，０００
ＳＲＶ０２	１，０２４	２０，０００

【図５】

テーブル１２３

ジョブ名称	消費予定メモリ容量（ＭＢ）	消費予定ディスク容量（ＭＢ）	スキップリミット
ＪＯＢ－Ａ	１８０	５０＋２．２×入力データサイズ	５
ＪＯＢ－Ｂ	５	３０	２

【図３】



【図６】

テーブル１７１

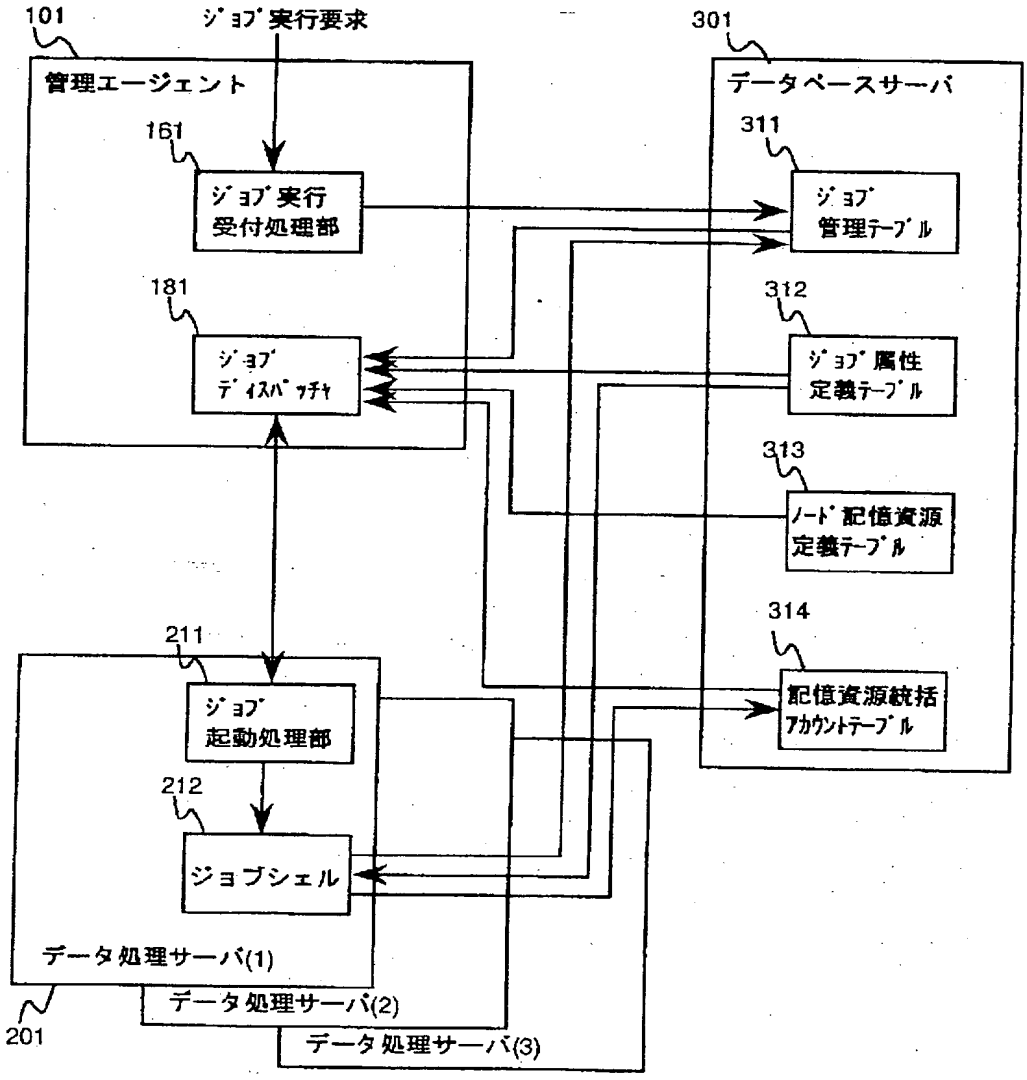
ジョブＩＤ	ジョブ名称	消費予定メモリ（ＭＢ）	消費予定ディスク容量（ＭＢ）	スキップカウンタ	スキップカウンタ	ジョブ状態
Ａ０１	JOB-A	１８０	１１，０５０	５	５	起動待ち
Ｂ０１	JOB-B	５	３０	０	３	実行中
Ａ０１	JOB-B	１８０	３，８００	１	５	起動待ち

【図９】

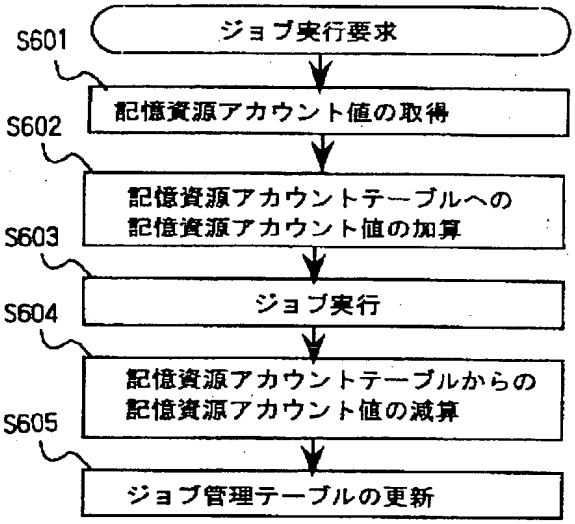
テーブル１４１

ノードＩＤ	消費予定メモリ容量（ＭＢ）	消費予定ディスク容量（ＭＢ）
ＳＲＶ０１	８００	２５，０００
ＳＲＶ０２	５００	１５，０００

【図１０】



【図１１】



フロントページの続き

(72)発明者　田中　紀夫
茨城県日立市大みか町五丁目２番１号　株
式会社日立製作所大みか工場内